

Timed trajectory generation for a toy-like wheeled robot

Jorge Bruno Silva

Industrial Electronic Department, University of Minho
Guimares, Portugal
jbruno@dei.uminho.pt

Vitor Matos

University of Minho
Guimares, Portugal
vmatos@dei.uminho.pt

Cristina Santos

University of Minho
Guimares, Portugal
cristina@dei.uminho.pt

Abstract—In this work, we address temporal stabilization of generated movements in autonomous robotics. We focus on generating movement for a mobile robot, that must reach a target location within a constant time. Target location is online calculated by using the robot visual system, such that action is steered by the sensory information. This is a very critical issue in several robotic tasks including: catching, hitting, and human-robot scenarios.

Robot velocity is controlled through an Hopf oscillator, adapted according to temporal feedback. Timing of the velocity profile is modulated according to an adaptive mechanism that enables setting different times for acceleration and deceleration.

Results on a DRK8000 mobile robot confirm the system's reliability with low-level sensors.

Keywords: *Timing, Nonlinear dynamical systems, Vision*

I. INTRODUCTION

Trajectory modulation and generation are two tightly coupled issues in robotics which have not been completely solved despite extensive research. These are fundamental issues in the path planning problem in autonomous robotics specially when operating in dynamic, partially unknown environments where a continuous online coupling to sensory information is required.

When considering common robotic tasks such as avoidance of moving objects; catching; hitting; juggling; coordination between degrees of freedom; human-robot scenarios and generating sequentially structured actions, temporal stabilization of movements, particularly in discrete movements, becomes relevant as well. These tasks may also include external temporal boundary conditions, which elicit and drive movement generation. Temporal stabilization means to keep as much as possible the planned movement time (MT), despite varying environmental conditions or perturbations. Movement must be compensated for when these disturbances either hold up, accelerate or decelerate it.

We have recently published [1] a controller inspired in the vertebrate biological motor systems [2], [3], [4], [5], structured in two functional hierarchical layers according to their level of abstraction. Each layer was formulated by autonomous nonlinear dynamical systems, due to their intrinsic properties: (1) low computational cost; (2) smooth online modulation while keeping the general features of the original movements;

(3) robustness against small perturbations; (4) and allows integration of sensory feedback [6], [7].

The lower level of the proposed controller acts out at the level of heading direction and velocity by formulating two separate dynamical systems. An attractor-based dynamics is formulated for the heading direction variable [8]. The velocity dynamics is based on past work for modeling a CPG [7], [9], [6]: a unique set of (oscillator-based) differential equations is able to produce complex movements modeled as periodic movements around time-varying offsets. The generated velocity profile and the switch between movement primitives can be easily modulated according to changes in values of the dynamical parameters.

The second level is responsible for selecting the most adequate motor behavior; map it onto the corresponding set of dynamical parameters and send them to the lower level at the right timing. A neural competitive dynamics models this second level. Its outcome are neural variables whose states encode the adequate set of parameters for the corresponding behavior. This switching dynamics autonomously bifurcate among possible behaviors, when relatively low level, noisy sensory information is used to initiate and steer action.

This timing problem had already been addressed using the dynamical systems theory. The framework proposed in [10] extends the attractor dynamics approach of behavior generation to the timing domain and is applied in [9] to an autonomous vehicle. However, [11] argues that the included temporal stabilization mechanism does not guarantee invariant movement time, and propose some changes including: (1) to use the dynamic solution to directly control the robot's velocity; (2) to use a full oscillator cycle; and (3) propose an adaptive rule that theoretically warrants invariant movement time.

In [1], timing of the velocity profile is modulated according to an adaptive mechanism that enables setting different times for acceleration and deceleration. Further, we applied bifurcation theory to switch the qualitative dynamics of the Hopf nonlinear system, instead of switching among different dynamical systems.

In this contribution, we close the sensory loop by online acquiring the target location through the robot visual system,

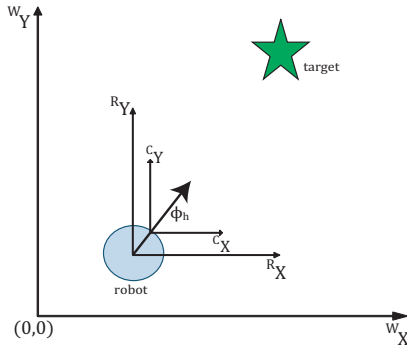


Fig. 1. Robot moving in an environment. The robot starts at the origin of the allocentric reference frame and moves towards the target. The robot has a camera mounted on top of it with its optical axis lying along the robot's heading direction.

such that action is steered by sensory information. We apply the Continuously Adaptive Mean Shift (CAMSHIFT) [12], due to its inherent advantages, such as its low computational cost and robustness. These properties enable the algorithm to be a color based real-time tracker, currently applied in computer-vision application problems.

The approach is demonstrated in a robot that navigates towards a target in a world and is confronted with unexpected disturbances, such as obstacles or sudden movements of the target. The robot is expected to deal with these disturbances and reach the target still respecting the desired movement time. Results in a DRK8000 lower level robot, confirm the system's reliability with low-level sensors and modest computational resources.

II. OVERALL ARCHITECTURE

In our application, a robot should reach a visually acquired target within a specified, constant movement time, independently of the environment configuration or perturbations. Fig. 1 depicts the robot scenario.

For generating these timed trajectories, we propose a system which is divided hierarchically in two functional levels according to their level of abstraction. Both of the levels are implemented through sets of interacting dynamical systems. The overall architecture is depicted in fig. 2.

We follow a modular approach by assuming that complex movement can be generated from the combination of simpler motor primitives, discrete and rhythmic, implemented as dynamical systems. This modularity is also assumed in terms of motor behaviors, stored as motor programs in the nervous system [13]. These assumptions turn a possibly high dimensional trajectory generation problem into a simple selection between pre-defined behaviors.

The first level is composed by the dynamics of two behavioral variables that control the 2D motion of the robot, its heading direction ϕ_h and forward velocity v . The directional dynamics governs the heading direction according to: the perceived target position as an attractor, and repellers erected

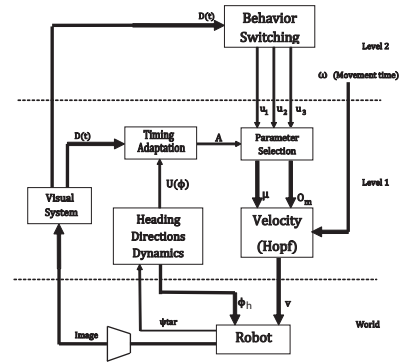


Fig. 2. System overall architecture.

by the detection of obstacles; safely steering the robot around obstacles into the target.

The velocity v of the robot is the major determinant to the success of the movement task within the specified timing constraints. Its value is the solution of a stable limit-cycle generated by a Hopf oscillator. The performed velocity consists of a single oscillation cycle, adapted in order to accelerate or decelerate in face of disturbances.

Commands are sent from the second level, changing qualitatively the dynamic solution by taking advantage of the Hopf bifurcation, resulting in three motor behaviors, available in the level's repertoire:

- 1) Stop: The robot does not move, as it awaits for the beginning of the task or has reached the target.
- 2) Execution: The robot performs the movement task. It starts and executes the pursuit of the target, avoiding eventual obstacles, while adapting its velocity to maintain the task's timing.
- 3) Rescue: In the eventual case of the target becoming out of reach within the time constraint due to an excessive disturbance, the robot maintains a constant, stable velocity to the target.

The second level selects and sequences the most appropriate motor behaviors accordingly to external conditions and current states of the task at hand.

In the following, we detail each module of the overall architecture.

III. ROBOT VISUAL SYSTEM

In this particular application the aim is to robustly detect a target (purple ball) in an unstructured environment. Target position is acquired by a camera mounted on the top of the robot and facing in the direction of the driving speed. We have assumed that target size is known and can be measured in the image.

However, in our application, we have to deal with the following main computer-vision problems such as: (1) clutter environment, (2) irregular object motion, (3) image noise and (4) high processing time. Since the application needs a fast algorithm for color tracking, we have adopt a color based real-time tracker called Continuously Adaptive Mean Shift

(CAMSHIFT) [12]. This algorithm deals with the described computer-vision application problems during its operation and has low cost computational.

It uses the color space HSV, which is less sensitive to lighting changes. The color model is mainly based on the hue histogram. The CAMSHIFT detects color objects using a probability distribution image of the desired color. It provides the image coordinates in pixels and the area of the color blob representing the detected object. To convert the blob coordinates in pixels to camera coordinates, the camera Pinhole model was used.

IV. HEADING DIRECTION CONTROL

The robot's heading direction, ϕ_h , in angular space and in an allocentric coordinate, is controlled by a nonlinear vector field in which task constraints contribute independently. The task of reaching the target, $F_{tar}(\phi_h)$, attracts ϕ_h towards the direction in which the target lies. The task of avoiding obstacles, $F_{obs}(\phi_h)$, repels ϕ_h from the direction in which obstacles are perceived.

Integration of these tasks is achieved by adding each of them to the vector field that governs heading direction dynamics.

$$\dot{\phi}_h = F_{obs}(\phi_h) + F_{tar}(\phi_h) + F_{stoch}. \quad (1)$$

A stochastic component, F_{stoch} , is added to ensure an escape from unstable states. For a full discussion see [14], [8], [9] for examples.

This approach differs from the potential field approach basically in the aspect that the state of the behavioral system must be in or near an attractor state of the dynamical system during operation.

V. VELOCITY CONTROL

For controlling the robot velocity we use the m variable of a nonlinear dynamical oscillator as follows:

$$\dot{m} = \alpha(\mu - r^2)(m - O_m) - \omega n, \quad (2)$$

$$\dot{n} = \alpha(\mu - r^2)n + \omega(m - O_m), \quad (3)$$

$$r = \sqrt{(m - O_m)^2 + n^2}, \quad (4)$$

where m and n are the state variables, amplitude of the oscillations is given by $A = \sqrt{\mu}$ for $\mu > 0$, ω specifies the oscillations frequency (rads⁻¹) and O_m is used to control the m solution offset.

This oscillator contains an Hopf bifurcation from a stable fixed point at $(m, n) = (O_m, 0)$ (when $\mu < 0$) to a structurally stable, harmonic limit cycle, for $\mu > 0$. Relaxation to these solutions is given by $\frac{1}{2\alpha\mu}$ (s).

This system is able to generate: (1) a discrete movement with offset O_m , if $\mu < 0$; (2) a rhythmic movement around O_m , if $\mu > 0$; and (3) the superimposition of both movements, resulting in a complex movement, if $\mu > 0$ and the offset is defined as a time-changing variable [6].

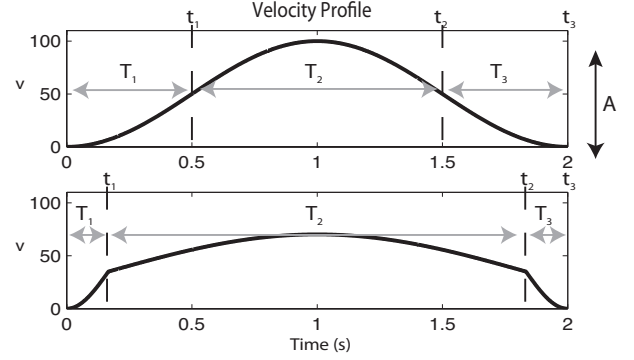


Fig. 3. Top: simple oscillation profile generated by the Hopf oscillator. On this profile, T_1 and T_2 are longer, resulting in a curve with a higher top velocity. Bottom: modulated oscillation profile where the acceleration and deceleration times are smaller, resulting in a smaller top velocity.

A. Profile modulation

The oscillator described by eqs. (2)(3)(4), generates a m harmonic solution which period is equal to the movement time $MT = \frac{2\pi}{\omega}$. MT is the time that velocity goes from zero to twice the oscillator radius, A , and back to zero again, executing a full sinusoidal cycle (fig. 3 (top)). However, this velocity profile exhibits an ascending and descending parts of the oscillatory cycle with equal durations. Thus a large amount of time is spent accelerating towards the maximum velocity and decelerating back to zero again. From a robot physical perspective, it is advantageous to keep the profile approximately constant as long as possible and minor top velocities being requested. Therefore, ideally it should be possible to choose the acceleration/deceleration time durations within the task's movement time.

Consider the velocity profile depicted in fig. 3 (top). The velocity evolves as follows:

$$v(t) = A(1 - \cos(\omega t)) \quad (5)$$

Considering the task of reaching a target, but subdividing the required velocity profile in three time intervals, each one with different durations, such their sum equals the movement time $MT = T_1 + T_2 + T_3$. For $t < t_1$, the oscillator covers the first quarter of the limit cycle (T_1), half of the limit cycle is covered for $t_1 < t < t_2$ (T_2), and the last quarter is covered from $t_2 < t < t_3$ (T_3).

For each of the three time intervals we calculate the angular frequency ω such that in the overall they are performed within the correct timing, as follows:

$$\omega_1 = \frac{\pi}{2T_1}, \quad \omega_2 = \frac{\pi}{T_2}, \quad \omega_3 = \frac{\pi}{2T_3}. \quad (6)$$

By integrating eq. (5) during the given time intervals, provides the distance s covered during each time interval, for a fixed radius cycle A :

$$s_1 = \frac{A(\frac{\pi}{2} - 1)}{\omega_1}, \quad s_2 = \frac{A(\pi - 2)}{\omega_2}, \quad s_3 = \frac{A(\frac{\pi}{2} - 1)}{\omega_3}. \quad (7)$$

The distance covered by the robot must be the distance needed to reach the target.

$$D(t=0) = s_1 + s_2 + s_3, \quad (8)$$

where $D(t=0)$ is the distance between the robot's initial position and the target coordinates at instant $t=0$ s.

B. Timing adaptation

In a real implementation disturbances of many types may occur which disturb the planned robot time course. [11] proposed an online adaptation rule for the Hopf radius which was able to guarantee that the remaining distance is traversed within the remaining time.

Building on the same formulation, we integrate eq. (5) during each of the three time intervals, determining an online updating rule for the Hopf offset and radius cycle A , that considers the current distance to the target and the remaining time to cover it. Depending on the current instant of time, the adaptive rule is calculated by different formulae as follows: For $0 < t < t_1$:

$$A_1(t) = \frac{D(t)}{\frac{\frac{\pi}{2}-1+\sin(\omega_1 t)}{\omega_1} + \frac{\pi+2}{\omega_2} + \frac{\frac{\pi}{2}-1}{\omega_3} - t} \quad (9)$$

For $t_1 < t < t_2$:

$$A_2(t) = \frac{D(t)}{\frac{\frac{\pi}{2}}{\omega_1} + \frac{\pi+1+\cos(\omega_2(t-T_1))}{\omega_2} + \frac{\frac{\pi}{2}-1}{\omega_3} - t} \quad (10)$$

For $t_2 < t < t_3$:

$$A_3(t) = \frac{D(t)}{\frac{\frac{\pi}{2}}{\omega_1} + \frac{\pi}{\omega_2} + \frac{\frac{\pi}{2}-\cos(\omega_3(t-T_1-T_2))}{\omega_3} - t} \quad (11)$$

C. Modulation of frequency and amplitude

By simply modifying both A and ω parameters, we modulate the velocity profile in amplitude and frequency respectively. These parameters are changed according to the oscillator current state, as follows:

$$A = \frac{A_1}{(1+e^{b(m-A)})(1+e^{bn})} + \frac{A_2}{1+e^{-b(m-A)}} + \frac{A_3}{(1+e^{b(m-A)})(1+e^{-bn})} \quad (12)$$

where A_1 , A_2 and A_3 are as defined in eq.(9-11).

Depending on the current values of the m and n variables, A alternate between three different values, A_1 , A_2 and A_3 . b controls the alternation speed between these values. The same procedure is used for the ω parameters.

D. Dealing with obstacles

An obstacle in the robot path obliges the robot to modify its path. However, robot velocity should be adjusted depending on the distance to the obstacle [8]: the closer the slower the robot moves. Thus, if an obstacle modifies the robot path, the corresponding change of timing has to be compensated.

The presence of obstacles is indicated by a potential function, $U(\phi_h)$ (see [8], [14] for details): if $U(\phi_h) < 0$, the

repulsion from obstacles contribution is weak for the current heading direction value; if $U(\phi_h) > 0$, the current heading direction ϕ_h , is on a repulsion zone of sufficient strength and the robot must change its heading direction in order to avoid the obstacle. In this situation the velocity should decrease by reducing A . This is achieved by modifying eq.(13) as follows:

$$A = (\text{eq.13}) \left(1 - \frac{TF-d}{1+e^{-b(U(\phi)-1/b)}} \right) \quad (13)$$

where d is the minimum distance to an obstacle measured by any of the robot's sensors and TF is the range of each sensor.

This group of equations constitutes the lower level responsible for setting the robot's velocity at each time step, according to the following set of parameters:

- 1) μ , switches on/off the rhythmic output. If $\mu > 0$ it also encodes the amplitude of rhythmic activity, $A = \sqrt{\mu}$;
- 2) O_m , modulates the oscillations' offsets, i.e. the goal for the discrete movement;
- 3) T_1, T_2, T_3 , specifies the velocity profile's shape.

The parameters α and b are set a priori.

This level receives from higher levels sets of parameters that specify and modulate in a simple and direct manner the generated trajectories.

VI. BEHAVIOR SWITCHING

In this work, there are three possible different behaviors: *stop*; *execution*; and *rescue*. Further, the switch between these behaviors should be easily and autonomously elicited, according to sensory information, such that action itself is elicited by perception. Each motor behavior will contribute with a value to the vector field. However, we do not intend that all motor behaviors are active simultaneously, but rather we want that only one is active at a time, while the other are disable.

This switching mechanism with such characteristics is implemented by a competitive dynamical system. It provides for stability against temporary fluctuations of the input signals and an autonomous switch easily elicited by the environment.

A "neural" variable $u_i \in [-1, 1]$ ($i = \text{stop, execution, rescue}$), represents each of the possible behaviors. A competitive dynamics is formulated for these variables as follows:

$$\alpha_\mu \dot{u}_i = \beta_i u_i - |\beta_i| u_i^3 - v \sum_{a \neq i} u_a^2 u_i + gwn \quad (14)$$

where neurons u_i can go "on" (=1) or "off" (=0). The neuron u_i , with the largest competitive advantage, $\beta_i > 0$, is likely to win the competition. The system is, however, multistable since for sufficiently small differences between the different β_i values multiple outcomes are possible.

α_μ defines the time scale of the dynamics. Parameter v is a competitive term which destabilizes any attractors in which more than one neuron is "on". Herein, is set to a constant value.

A sequence of neural switches and hence behavior switching, is generated by translating sensory conditions and logical constraints into values for the β_i parameters ([15], [10], [9], [16] for examples). $\beta_i \in [-1, 1]$ and is achieved by setting:

$\beta_i = 1.5 + 2b_i$, where b_i are “quasi - boolean” variables, varying between 0 and 1 (with a tendency to have values either close to 0 or close to 1).

Neuron *execution* is “on”(=1) when: (1) $t > t_{\text{init}}$; **and** (2) target is reachable; **and** (3) target has not been reached. A target is considered reachable when it is possible to reach the target in the remaining time.

Neuron *stop* is “on”(=1) when one of the following is true: (1) time, t , is below the initial time, t_{init} , set by user; (2) target has been reached.

Similarly, neuron *rescue* is “on”(=1) when: (1) $t > t_{\text{init}}$; **and** (2) target is not reachable.

A. Parameter modulation

Different values of triplets of neurons (u_{stop} , $u_{\text{execution}}$, u_{rescue}) lead to different behaviors, namely: no movement, timed movement and constant movement. Each triplet must then be mapped onto different values for the set of parameters, modulating the Hopf oscillator such that the desired robot motion is achieved. This is achieved as follows.

1) *Offset* O_m : The offset O_m is modulated depending on the neurons values as follows:

$$O_m = |u_{\text{stop}}| \times 0 + |u_{\text{execution}}| \times A + |u_{\text{rescue}}| \times 0.1, \quad (15)$$

2) *Oscillatory Activity*: Qualitatively, by modifying on the fly the μ parameter, the system switches between a stable fixed point at $m = O_m$ (for $\mu < 0$) and a purely rhythmic movement (for $\mu > 0$). Hence, the μ parameter controls whether or not there are oscillations and thus, timed movement.

For $\mu > 0$ this parameter also encodes the amplitude of rhythmic activity. This parameter depends on the neural competitive dynamics as follows:

$$\mu = -(|u_{\text{stop}}| + |u_{\text{rescue}}|) \frac{A^2}{2} + |u_{\text{execution}}| A^2 \quad (16)$$

This means that the timed movement is performed when $u_{\text{execution}}$ is “on”, by making $\mu > 0$, and that the velocity is constant when u_{stop} or u_{rescue} are “on”, by making $\mu < 0$.

VII. EXPERIMENTAL RESULTS

In this section we describe some experiments done in the *DRK8000* mobile robot. Initially the robot is stopped and it must reach a target in two different world configurations within a specified time. The robot has no previous knowledge of any of the obstacles in the world, or of any other disturbance that may occur. Targets are acquired by the robot visual system.

At each sensorial cycle, sensory information is acquired, dynamic equations are calculated and integrated using an Euler method with time step of 90 ms. The maximum robot velocity is 0.15 m/s. A movement time of 30s is specified. During the first $t_{\text{init}} = 3s$, the robot only turns towards the target but no timed forward movement is generated.

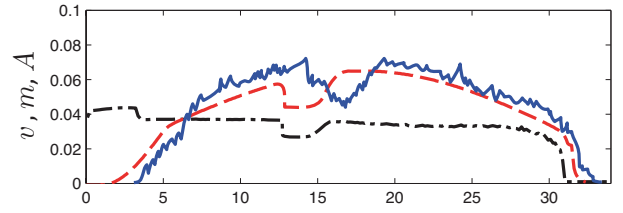


Fig. 4. Robot’s real velocity (blue continuous line), the timing velocity (red dashed line) and the amplitude of the oscillator (black mixed line). The ball is placed 2m away from the robot. Obstacles are 1m away from the robot.

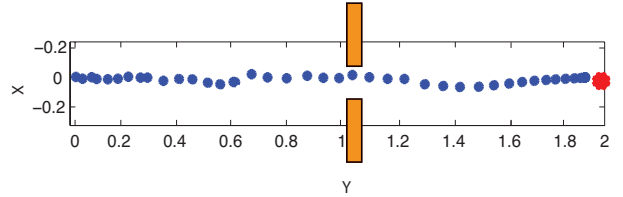


Fig. 5. The robot’s path as recorded by the dead-reckoning for the situation depicted in experiment 1. The red circle indicates the target location and the rectangles indicate the obstacle positions in the world.

A. Experiment 1

In this experiment the robot is faced with two obstacles during its path towards a target positioned at 2m away. Both obstacles are at 1 m from the robot’s initial position, but separated such that the robot is able to pass in between. Despite this disturbance, we expect that the robot will avoid the obstacles and reach the target around the specified movement time, 30 s.

The velocity profile of this experiment is presented on fig. 4. Fig. 5 depicts the robot path throughout the experiment. Markers depict robot position at instants of time, thus giving an indication of the robot velocity. At $t = 13$ s, when the robot is close to the obstacles, the amplitude of oscillator is decreased according to eq. (13), consequently reducing the robot’s velocity. A safe passage between the two obstacles is performed. After passing between the obstacles ($t = 16$ s), the amplitude is raised in order to compensate for the provoked delay.

B. Experiment 2

The purpose of this experiment illustrated in fig.6, is to show that the robot also compensates target displacement during its motion. Fig. 7 (bottom), presents the robot distance to the target during the experiment.

Initially, the robot tries to reach within the 30 s, a target positioned at 2 m. However, at $t = 19s$ (panel 2 in fig.6), the target is displaced to a distance of 1.4 m relatively the robot’s initial position.

In fig.7 (top) it is noticeable the simultaneous adaptation of the oscillator solution and the real velocity of the robot (blue continuous line) at the moment of displacement.

Observing table VII-B we can see that independently of the complexity of the world and the disturbances that the robot

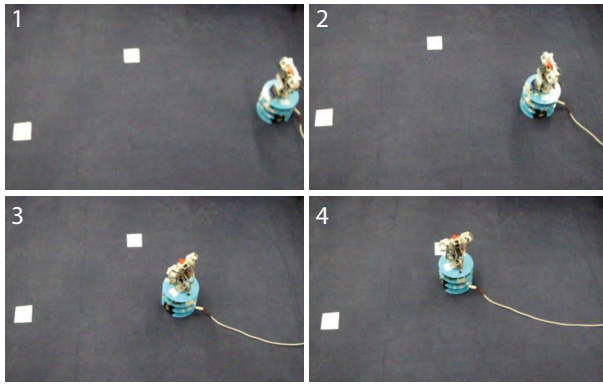


Fig. 6. Robot motion when the robot is moving to a target and suddenly at $t = 19$ s the target is displaced (panel B). Robot initial position is depicted with a cross. Robot path is show by a dashed yellow trace.

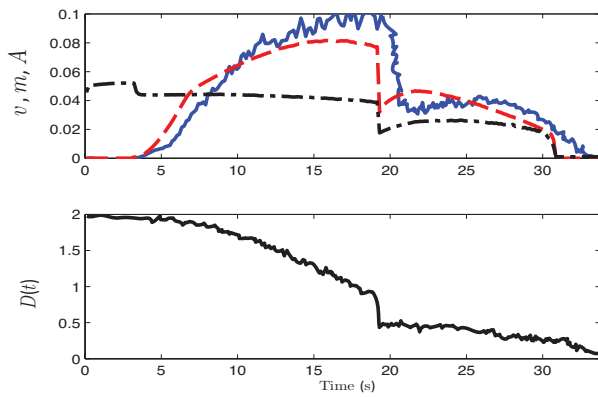


Fig. 7. Top: robot's real velocity (blue continuous line), the timing velocity (red dashed line) and the amplitude of the oscillator (black mixed line). Bottom: distance between the target and the robot along all trajectory.

faces (obeying the robot's physical restrictions), the robot performs its movement task within the specified movement time.

VIII. CONCLUSION

In this paper we addressed the problem of generating timed trajectories for autonomous vehicles equipped with both modest computational resources and noisy, low-level sensors. The robot is expected to reach a target within a stipulated time independently of the complexity of the environment and distance to the target, being able to detect and circumnavigate any obstacle and moreover compensate any delay or rush. Further, the robot online calculates the target position, through a camera mounted on its top, such that action is steered by the noisy sensory information.

TABLE I
AVERAGE PERFORMED MOVEMENT TIME IN DIFFERENT WORLD CONFIGURATIONS

Experiments	Time (s)	MT (s)	Initial Distance (m)
Experiment 1	29.2	30	2
Experiment 2	29.4	30	2

The system was able to: 1) initiate and terminate the movement task through the exploitation of the oscillator's Hopf bifurcation; 2) adapt the velocity profile, allowing to specify the duration of specific parts, 3) adapt the timing of the generated trajectories. The proposed system is able to adapt the robot velocity in order to safely overcome obstacles.

We successfully demonstrated the reliability of the proposed system through two experiments in a *DRK8000* robot. Future work will address the ability to endow the system with cognitive capabilities considering targets that may disappear during short periods of time, prediction and forgetting.

ACKNOWLEDGMENT

Work supported by the Portuguese Science Foundation (grant PTDC/EEA-CRO/100655/2008).

REFERENCES

- [1] C. Santos and M. Ferreira. Generating trajectories with temporal constraints for an autonomous robot. In *SSRR*, 2010.
- [2] S. Grillner. Locomotion in vertebrates: central mechanisms and reflex interaction. *Physiological Reviews*, 55:247–304, 1975.
- [3] M. MacKay-Lyons. Central pattern generation of locomotion: a review of the evidence. *Phys Ther*, 82(1):69–83, January 2002.
- [4] S. Grillner, P. Wallna, K. Saitoha, A. Kozlova, and B. Robertson. Neural bases of goal-directed locomotion in vertebrates: An overview. *Brain Research Reviews*, 57(1):2–12, January 2008.
- [5] S. Grillner, J. Hellgren, A. Ménard, K. Saitoh, and M. A. Wikström. Mechanisms for selection of basic motor programs—roles for the striatum and pallidum. *Trends Neurosci*, 28(7):364–370, July 2005.
- [6] S. Degallier, C. Santos, L. Righetti, and A. Ijspeert. Movement generation using dynamical systems: a humanoid robot performing a drumming task. In *IEEE-RAS International Conference on Humanoid Robots*, 2006.
- [7] L. Castro, C. Santos, M. Oliveira, and A. Ijspeert. Postural control on a quadruped robot using lateral tilt: A dynamical system approach. In *EUROS*, volume 44 of *Springer Tracts in Advanced Robotics*, pages 205–214. Springer, 2008.
- [8] E. Bicho, P. Mallet, and G. Schöner. Target representation on an autonomous vehicle with low-level sensors. *The International Journal of Robotics Research*, (210):424–447, 2000.
- [9] C. Santos. Generating timed trajectories for an autonomous vehicle: a non-linear dynamical systems approach. in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004.
- [10] G. Schöner and C. Santos. Control of movement time and sequential action through attractor dynamics: A simulation study demonstrating object interception and coordination. in *Proc. of the 9th Int. Symposium on Intelligent Robotic Systems (SIRS)*, 2001.
- [11] J. Lipinski, Y. Sandamirskaya, and G. Schöner. Swing it to the left, swing it to the right: Enacting flexible spatial language using a neurodynamic framework. *Cognitive Neurodynamics, special issue on "Language Dynamics"*, 3(4), 2009.
- [12] E. Large. Computer vision face tracking as a component of a perceptual user interface. in *Workshop on Applications of Computer Vision, Princeton, NJ*, pages 214–219, 1998.
- [13] S. Grillner, Ö. Ekeberg, A. El Manira, A. Lansner, D. Parker, J. Tegner, and P. Wallen. Intrinsic function of a neuronal network – a vertebrate central pattern generator. *Brain Research Reviews*, 1998.
- [14] G. Schöner and M. Dose. A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion. *Robotics and Autonomous Systems*, 10(4):253–267, 1992.
- [15] A. Steinhage and G. Schöner. Dynamical systems for the behavioral organization of autonomous robot navigation. In *McKee G T Schenker PS, editor, Sensor Fusion and Decentralized Control in Robotic Systems: Proceedings of Spie-Intelligent Systems Manufacturers, Boston*, pages 169–180, 1998.
- [16] C. Santos and M. Ferreira. Two vision-guided vehicles: temporal coordination using nonlinear dynamical systems. In *ICRA*, pages 14–19, 2007.